Topologische Klassifikation kubischer Flächen in $\mathbb{P}^3(\mathbb{R})$ anhand von Julia-Implementierungen: 5 Fallbeispiele

Yimu Mao

Zusammenfassung

Deutsch: Dieser Artikel zeigt, wie mit der Programmiersprache Julia und computeralgebraischen Bibliotheken eine automatische Klassifikation der topologischen Typen reeller kubischer Flächen im projektiven Raum durchgeführt werden kann. Die Analyse basiert auf vorgegebenen reellen kubischen Flächen. Der Code wurde von mir auf Basis von `https://mathrepo.mis.mpg.de/27pAdicLines/JuliaCode.html` erweitert. Das Projekt entstand im Praktikum am Institut für Geometrie der TU Dresden (23. Juni – 4. Juli 2025).

Es wird gezeigt, wie Singularitäten berechnet, Glattheit geprüft und Zusammenhangskomponenten gezählt werden. Fünf Beispiele veranschaulichen die topologischen Unterschiede.

Bis auf die geprüften linearen Transformationen kann für die Ergebnisse keine vollständige Korrektheit garantiert werden. Alle Arbeiten erfolgten eigenständig mit Open-Source-Software. Der Code dient als Referenz für Lehre und Forschung.

Besonderer Dank gilt Professor Mario Kummer für seine Unterstützung.

English: This article shows how Julia and computer algebra tools classify topological types of real cubic surfaces in projective space. The code extends `https://mathrepo.mis.mpg.de/27pAdicLines/JuliaCode.html` and was developed during an internship at the Institute of Geometry, TU Dresden (June 23 – July 4, 2025).

It includes singularity computation, smoothness checks, and counting connected components, with five examples.

Complete correctness isn't guaranteed beyond verified linear transformations. Work was done independently using open-source software. The code supports teaching and research.

Thanks to Professor Mario Kummer for his support.

Русский: В этой статье показано, как с помощью языка программирования `Julia` и библиотек компьютерной алгебры можно классифицировать топологические типы действительных кубических поверхностей в проективном пространстве. Код является расширением `https://mathrepo.mis.mpg.de/27pAdicLines/JuliaCode.html` и был разработан во время стажировки в Техническом университете Дрездена (23 июня – 4 июля 2025 года).

В работе рассматриваются вычисление особенностей, проверка гладкости и подсчёт связных компонентов на примере пяти случаев.

Полная корректность результатов, кроме проверенных линейных преобразований, не гарантируется. Работа выполнена самостоятельно с использованием программного обеспечения с открытым исходным кодом. Код предназначен для использования в преподавании и научных исследованиях.

Особая благодарность профессору Марио Куммеру за поддержку.

中文：这篇文章展示了如何使用Julia语言和计算机代数工具对射影空间中实三次曲面的拓扑类型进行分类。代码基于`https://mathrepo.mis.mpg.de/27pAdicLines/JuliaCode.html`进行了扩展，并于2025年6月23日至7月4日在德累斯顿工业大学几何研究院实习期间开发完成。

内容包括奇点计算、光滑性检测以及连通分支计数，并通过五个示例加以说明。

除已验证的线性变换外，结果的完全正确性无法保证。所有工作均独立完成，采用开源软件。代码可用于教学和科研参考。

特别感谢Mario Kummer教授的支持。

Inhaltsverzeichnis

1 Fragebeschreibung

Ich habe folgend kubische homogenes Polynom, und ich muss den Julia-Code im angegebenen Link verwenden, um die topologische Struktur zu bestimmen, die diesen kubischen homogenes Polynom entspricht

1. $F_1(x,y,z,w) = 72*x^3+1152*x^2*y-96*x*y^2-1664*y^3+3*x^2*z-90*x*y*z-408*y^2*z+9*x*z^2+32*y*z^2+300*x^2*w-1368*x*y*w-3008*y^2*w-382*y*z*w+23*z^2*w-408*x*w^2-1480*y*w^2-67*z*w^2-220*w^3$

2. $F_2(x,y,z,w) = 2052*x^3-162*x^2*y+1800*x*y^2-3680*y^3+2853*x^2*z-1476*x*y*z-4964*y^2*z-3114*x*z^2+376*y*z^2+645*z^3+1890*x^2*w+3768*x*y*w-9976*y^2*w-2580*x*z*w-4728*y*z*w+664*z^2*w+444*x*w^2-5638*y*w^2-793*z*w^2-866*w^3$

3. $F_3(x,y,z,w) = 621*x^3+2619*x^2*y-1800*x*y^2-3040*y^3+315*x^2*z+618*x*y*z-1212*y^2*z+174*x*z^2+172*y*z^2+1152*x^2*w-24*x*y*w-4328*y^2*w+306*x*z*w-650*y*z*w+38*z^2*w+51*x*w^2-1907*y*w^2-97*z*w^2-256*w^3$

Ich muss zunächst feststellen, ob dieses Polynom smooth (glatt) ist oder nicht, eine kubische Fläche ist nur dann eine del Pezzo-Fläche, wenn sie smooth (glatt) ist. z.B

$$F_1 = 0, \frac{\partial F_1}{\partial x} = 0, \frac{\partial F_1}{\partial y} = 0, \frac{\partial F_1}{\partial z} = 0, \frac{\partial F_1}{\partial w} = 0$$

Wenn dieses Gleichungssystem keine reelle Lösung auf $\mathbb{P}^3(\mathbb{R})$ hat, dann ist es F_1 smooth (glatt). Ich nehme mal an, dass diese alle drei Polynome smooth (glatt) sind. Glatt bedeutet in dem Zusammenhang von del Pezzo Flächen, dass gar keine singulären Punkte existieren - auch keine komplexen.

2 Homogenes Polynom F_1

Was ich schon bekommt habe sind folgend

```
  Tracking 81 paths... 100%|###############################| Time: 0:00:10
# paths tracked:                    81
# non-singular solutions (real):   27 (27)
# singular endpoints (real):       0 (0)
# total solutions (real):          27 (27)
```

Das ist schon ziemlich sehr klar, das bedeutet

- Es wurden insgesamt 81 Pfade verfolgt (entspricht der theoretischen Anzahl der Lösungen des Systems, einschließlich komplexer und reeller Lösungen), der Algorithmus wurde vollständig ausgeführt.
- 81 Pfade wurden verfolgt, wie oben angegeben.
- Es wurden 27 nicht-singuläre reelle Lösungen gefunden, und alle 27 sind nicht-singulär.
- Es gibt keine singulären reellen Lösungen
- Die Gesamtanzahl der reellen Lösungen beträgt 27

Von dem Corollary 3.3"des The antibirational involutions of the plane and the classification of real del Pezzo surfaces"[Rus02] können wir wissen, dass die topologische Struktur $\#7\mathbb{RP}^2$. Julia Code ist in den nächst Seite

```
1  using HomotopyContinuation
2  using Distributions
3  
4  HomotopyContinuation.@var(x,y,z,w)
5  HomotopyContinuation.@var(a,b,c,d)
6  
7  function fixedCubicSurface()
8      return 72*x^3 + 1152*x^2*y - 96*x*y^2 - 1664*y^3 +
9              3*x^2*z - 90*x*y*z - 408*y^2*z + 9*x*z^2 + 32*y*z^2 +
10             300*x^2*w - 1368*x*y*w - 3008*y^2*w - 382*y*z*w +
11             23*z^2*w - 408*x*w^2 - 1480*y*w^2 - 67*z*w^2 - 220*w^3
12 end
13 
14 function realLineCounts(lambda, M)
15     results = Dict([3=>0.0, 7=>0.0, 15=>0.0, 27=>0.0]);
16 
17     for i in 1:M
18         bool = true
19 
20         while bool
21 
22             f = fixedCubicSurface();
23 
24             g0 = subs(f, [x,y,z,w] => [a, c, 1, 0]);
25             g1 = subs(f, [x,y,z,w] => [a+b, c+d, 1, 1]);
26             g2 = subs(f, [x,y,z,w] => [a-b, c-d, 1, -1]);
27             g3 = subs(f, [x,y,z,w] => [b, d, 0, 1]);
28 
29             S = System([g0,g1,g2,g3]);
30             A = solve(S);
31 
32             r = size(real_solutions(A))[1];
33 
34             if !(r in [3,7,15,27])
35                 bool = true
36             else
37                 bool = false;
38                 results[r] += 1;
39             end;
40         end;
41     end;
42 
43     return results;
44 end;
45 
46 M = 10000
47 l = 1/100
48 results = realLineCounts(l, M);
49 println(results)
```

3 Homogenes Polynom F_2

Mit Gleich Methode können wir die folgend Ausgabe bekommen, nur die Reihe 8 bis 11 verändern

```
    Tracking 81 paths... 100%|##############################| Time: 0:00:04
  # paths tracked:                  81
  # non-singular solutions (real):  27 (27)
  # singular endpoints (real):      0 (0)
  # total solutions (real):         27 (27)
```

Das bedeutet,

- Es wurden insgesamt 81 Pfade verfolgt (entspricht der theoretischen Anzahl der Lösungen des Systems, einschließlich komplexer und reeller Lösungen), der Algorithmus wurde vollständig ausgeführt.
- 81 Pfade wurden verfolgt, wie oben angegeben.
- Es wurden 27 nicht-singuläre Lösungen gefunden, davon sind 27 reell.
- Es gibt keine singulären Endpunkte (Lösungen), weder reell noch komplex.
- Die Gesamtanzahl der reellen Lösungen beträgt 27 (davon 27 reell).

Nun können wir wissen, dass die topologische Struktur $\#7\mathbb{RP}^2$.

4 Homogenes Polynom F_3

```
  Tracking 81 paths... 100%|##############################| Time: 0:00:04
# paths tracked:                   81
# non-singular solutions (real):   27 (27)
# singular endpoints (real):       0 (0)
# total solutions (real):          27 (27)
```

Das bedeutet,

- Es wurden insgesamt 81 Pfade verfolgt (entspricht der theoretischen Anzahl der Lösungen des Systems, einschließlich komplexer und reeller Lösungen), der Algorithmus wurde vollständig ausgeführt.
- 81 Pfade wurden verfolgt, wie oben angegeben.
- Es wurden 27 nicht-singuläre Lösungen gefunden, davon sind 27 reell.
- Es gibt keine singulären Endpunkte (Lösungen), weder reell noch komplex.
- Die Gesamtanzahl der reellen Lösungen beträgt 27 (davon 27 reell).

Nun können wir wissen, dass die topologische Struktur $\#7\mathbb{RP}^2$.

5 Homogenes Polynom F_4

Diese alle sind $\#7\mathbb{RP}^2$. Diese sind nicht so charakteristisch, deswegen habe ich die unten Beispiel ausgedacht.

1. $F_4(x,y,z,w) = F_3(x,y,z,w) + 256*w^3 = 621*x^3 + 2619*x^2*y - 1800*x*y^2 - 3040*y^3 + 315*x^2*z + 618*x*y*z - 1212*y^2*z + 174*x*z^2 + 172*y*z^2 + 1152*x^2*w - 24*x*y*w - 4328*y^2*w + 306*x*z*w - 650*y*z*w + 38*z^2*w + 51*x*w^2 - 1907*y*w^2 - 97*z*w^2$

```
  Tracking 81 paths... 100%|##############################| Time: 0:00:010
# paths tracked:                   81
# non-singular solutions (real):   27 (3)
# singular endpoints (real):       0 (0)
# total solutions (real):          27 (3)
```

Das bedeutet,

- Es wurden insgesamt 81 Pfade verfolgt (entspricht der theoretischen Anzahl der Lösungen des Systems, einschließlich komplexer und reeller Lösungen), der Algorithmus wurde vollständig ausgeführt.

- 81 Pfade wurden verfolgt, wie oben angegeben.
- Es wurden 27 nicht-singuläre Lösungen gefunden, davon sind 3 reell.
- Es gibt keine singulären Endpunkte (Lösungen), weder reell noch komplex.
- Die Gesamtanzahl der reellen Lösungen beträgt 27 (davon 3 reell).

Nun können wir wissen, dass die topologische Struktur $\#\mathbb{RP}^2$ oder $\mathbb{RP}^2 \sqcup S^2$. ich werde dannach bestimmen, welsche es ist.

Um zu entscheiden, ob es sich um $\#\mathbb{RP}^2$ oder um $\mathbb{RP}^2 \sqcup S^2$ handelt, können wir versuchen zu überprüfen, ob die reellen Lösungen alle in einer Ebene liegen. Das bedeutet, dass die drei reellen Lösungen drei Geraden entsprechen. Wir ändern daher den Julia-Code so ab, dass er die Gleichungen dieser drei Geraden ausgibt, damit wir überprüfen können, ob sie in einer gemeinsamen Ebene liegen. Auch wenn wir aufgrund der begrenzten Genauigkeit von Computern nicht mit absoluter Sicherheit feststellen können, ob die Geraden koplanar sind, können wir zumindest beurteilen, ob sie windschief (d.h. nicht in einer Ebene) sind.

Auf $\mathbb{RP}^2 \sqcup S^2$ existieren in dem S^2-Teil keine reellen Geraden, daher können alle reellen Geraden nur im $\mathbb{RP}^2$-Teil liegen. Wenn wir also auf $\mathbb{RP}^2 \sqcup S^2$ drei verschiedene reelle Geraden finden, dann müssen sie alle im $\mathbb{RP}^2$-Teil liegen. Da $\mathbb{RP}^2$ eine ganz speziell an den Eigenschaften hat, liegen drei Geraden darin notwendigerweise in einer gemeinsamen Ebene.

Ich habe den Code geändert. Die Ausgabe sieht jetzt wie folgt aus:

```
Insgesamt 3 reelle Geraden gefunden.

Reelle Gerade Nr. 1:
 Gleichung 1: -0.213808519127661*w + 0.684791593501686*x
 + 0.345995122285053*y + 0.604676579646351*z = 0
 Gleichung 2: 0.808239218733016*w + 0.272505815030036*x
 + 0.442912421042277*y - 0.276258091941447*z = 0
--------

Reelle Gerade Nr. 2:
 Gleichung 1: -0.350317042954575*w + 0.0483254543498994*x
 + 0.86478485586579*y + 0.356496525849304*z = 0
 Gleichung 2: 0.812261897004688*w + 0.239542513259883*x
 + 0.439258341318511*y - 0.299836796603745*z = 0
--------

Reelle Gerade Nr. 3:
 Gleichung 1: -0.317313737336273*w - 0.764975611629068*x
 + 0.5481044807965*y + 0.117071703845205*z = 0
 Gleichung 2: 0.763278109248239*w + 0.0532094551211621*x
 + 0.577125063336332*y - 0.285485451638394*z = 0
--------
```

Dann bekommt man die Linear bekommen

$$\mathbf{X}_1 = r_1 \begin{pmatrix} -0.6245 \\ 0.2458 \\ 0.6750 \\ 0.3066 \end{pmatrix} + r_2 \begin{pmatrix} 0.2585 \\ -0.7897 \\ 0.3201 \\ 0.4550 \end{pmatrix}, \mathbf{X}_2 = s_1 \begin{pmatrix} -0.2684 \\ -0.3705 \\ 0.8879 \\ -0.0482 \end{pmatrix} + s_2 \begin{pmatrix} 0.9330 \\ -0.2169 \\ 0.1794 \\ -0.2241 \end{pmatrix}, \mathbf{X}_3 = t_1 \begin{pmatrix} -0.0069 \\ -0.2773 \\ 0.9497 \\ -0.1451 \end{pmatrix} + t_2 \begin{pmatrix} 0.6411 \\ -0.6093 \\ -0.1038 \\ 0.4548 \end{pmatrix}$$

Es ist leicht zu erkennen, dass sich diese drei Geraden nicht schneiden, daher liegen sie nicht in einer gemeinsamen Ebene, weshalb die topologische Struktur $\boxed{\#\mathbb{RP}^2}$ ist.

Ich habe den Julia-Code verändert, es sieht wie unten aus

```
using HomotopyContinuation
using LinearAlgebra

@var x y z w
@var a b c d

function festeKubischeFlaeche()
    return 621*x^3+2619*x^2*y-1800*x*y^2-3040*y^3+315*x^2*z+
           618*x*y*z-1212*y^2*z+174*x*z^2+172*y*z^2+1152*x^2*w-
           24*x*y*w-4328*y^2*w+306*x*z*w-650*y*z*w+38*z^2*w+
           51*x*w^2-1907*y*w^2-97*z*w^2
end

function realLineEquations()

    f = festeKubischeFlaeche()

    g0 = subs(f, [x,y,z,w] => [a, c, 1, 0])
    g1 = subs(f, [x,y,z,w] => [a + b, c + d, 1, 1])
    g2 = subs(f, [x,y,z,w] => [a - b, c - d, 1, -1])
    g3 = subs(f, [x,y,z,w] => [b, d, 0, 1])

    S = System([g0, g1, g2, g3])

    result = solve(S)

    vars = variables(S)

    real_sols = real_solutions(result)

    println("\n Insgesamt $(length(real_sols)) reelle Geraden gefunden.\n")

    for (index, sol) in enumerate(real_sols)

        a_val = sol[findfirst(==(a), vars)]
        b_val = sol[findfirst(==(b), vars)]
        c_val = sol[findfirst(==(c), vars)]
        d_val = sol[findfirst(==(d), vars)]

        P1 = [a_val, c_val, 1.0, 0.0]
        P2 = [a_val + b_val, c_val + d_val, 1.0, 1.0]

        V = hcat(P1, P2)
        N = nullspace(V')

        println("Reelle Gerade Nr. $(index):")

        for i in 1:size(N, 2)

            coeffs = N[:, i]
            equation = coeffs[1]*x + coeffs[2]*y + coeffs[3]*z + coeffs[4]*w
            println("  Gleichung $i: $equation = 0")

        end

        println("--------\n")

    end

end

realLineEquations()
```

Wir können ebenso den Rang der folgenden Matrix **B** berechnen. Der Rang dieser Matrix entspricht genau dem $\rho(\mathbf{X}_{\mathbb{R}})$ aus Korollar 3.3 im Buch The antibirational involutions of the plane and the classification of real del Pezzo surfaces"von Autor Francesco Russo [Rus02].

$$\begin{bmatrix} 3 & 1 & 1 & 1 \\ 1 & & & \\ 1 & & \mathbf{A} & \\ 1 & & & \end{bmatrix} \tag{1}$$

und $\mathbf{A} = (a_{ij})_{1\leq i,j\leq r}$ damit ist r die Anzahl von reelle Geraden $L_1, L_2, ...L_r$ Dann

$$\begin{cases} a_{ii} = -1 \\ a_{ij} = |L_i \cap L_j|, i \neq j \end{cases}$$

$$\mathbf{B}_1 = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix} \Longrightarrow \mathbf{B}_1 = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 0 & -\frac{4}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & -\frac{5}{4} & -\frac{1}{4} \\ 0 & 0 & 0 & -\frac{6}{5} \end{bmatrix} \tag{2}$$

deswegen $\operatorname{rank}(\mathbf{B}_1) = \rho(\mathbf{X}_{\mathbb{R}}) = 4$, dann ist es $\#\mathbb{RP}^2$

6 Homogenes Polynom F_5

F_5 ist eine Beispiel aus [Haku]

$$\begin{aligned} h &= x_0^3 - x_0(2x_1^2 + 2x_2^2 + x_3^2) + x_1^3 + x_1x_2^2 \\ &= x^3 - x(2y^2 + 2z^2 + w^2) + y^3 + yz^2 \end{aligned}$$

```
   Tracking 81 paths... 100%|###############################| Time: 0:00:04
  # paths tracked:                  81
  # non-singular solutions (real):  21 (3)
  # singular endpoints (real):      0 (0)
  # total solutions (real):         21 (3)

Insgesamt 3 reelle Geraden gefunden.

Reelle Gerade Nr. 1:
 Gleichung 1: -0.133808283664909*w - 0.965450849718747*x -
 0.223606797749979*y = 0
 Gleichung 2: 0.481762745781211*w + 0.133808283664909*x -
 0.866025403784439*y - 2.77555756156289e-17*z = 0
--------

Reelle Gerade Nr. 2:
 Gleichung 1: 0.133808283664909*w - 0.965450849718747*x -
 0.223606797749979*y = 0
 Gleichung 2: 0.481762745781211*w - 0.133808283664909*x +
 0.866025403784439*y + 2.77555756156289e-17*z = 0
--------

Reelle Gerade Nr. 3:
 Gleichung 1: -1.0*x = 0
 Gleichung 2: -1.0*y = 0
--------
```

Dann bekommt man die Linear bekommen

$$\mathbf{X}_1 = r_1 \begin{pmatrix} 0.0000 \\ 0.0000 \\ 1.0000 \\ 0.0000 \end{pmatrix} + r_2 \begin{pmatrix} -0.2236 \\ 0.4472 \\ 0.0000 \\ 0.8660 \end{pmatrix}, \mathbf{X}_2 = s_1 \begin{pmatrix} 0.0000 \\ -0.0000 \\ 1.0000 \\ -0.0000 \end{pmatrix} + s_2 \begin{pmatrix} 0.2236 \\ -0.4472 \\ 0.0000 \\ -0.8660 \end{pmatrix}, \mathbf{X}_3 = t_1 \begin{pmatrix} 0.0000 \\ 0.0000 \\ 1.0000 \\ 0.0000 \end{pmatrix} + t_2 \begin{pmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \\ 1.0000 \end{pmatrix}$$

Das ist sehr deutlich, dass diese drei Linear Schnittpunkt haben. deswegen ist es $\boxed{\mathbb{RP}^2 \sqcup \mathbf{S}^2}$

Wir können auch mit Matrix machen

$$\mathbf{B}_2 = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \Longrightarrow \mathbf{B}_2 = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 0 & -\frac{4}{3} & \frac{2}{3} & \frac{2}{3} \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

deswegen $\operatorname{rank}(\mathbf{B}_2) = \rho(\mathbf{X}_{\mathbb{R}}) = 3$, dann ist es $\mathbb{RP}^2 \sqcup \mathbf{S}^2$

Literatur

[Rus02] Francesco Russo, The antibirational involutions of the plane and the classification of real del Pezzo surfaces. In Algebraic geometry, pages 289–312. de Gruyter, Berlin, 2002

[Max Planck Institute for Mathematics in the Sciences] Julia Code for sampling real cubics, Max Planck Institute for Mathematics in the Sciences

[Haku] Christoph Hanselka and Mario Kummer, Positive ulrich sheaves.